Neural Networks for Single Image Super-Resolution (SISR)

Luisa Neubauer, Mikail Deniz Cayoglu

Abstract

This paper explores different architectures for the task of up-sampling LR (Low Resolution) images - also known as Single Image Super Resolution - in a meaningful way. It contrasts two classical interpolation algorithms (bicubic interpolation and nearest neighbor Interpolation) with neural network based methods. Specifically, the performance of the SRCNN and VDSR network architectures is investigated. Finally, a comparative analysis is made based on the two evaluation metrics SSIM and PSNR that quantify the reconstruction quality of the output images.

I. Introduction

IMAGE Super-Resolution - abbreviated as SR - is the process of up-sampling a Low-Resolution Image (LR) in a meaningful way to transform it into an output image of high(er) resolution. This task can be performed in a plethora of ways, from applying classical interpolation algorithms of different orders (such as bi-linear, bi-cubic or nearest neighbor interpolation) to training (adversarial) neural networks. The goal is to recover details from the rough contours of the low-resolution image, thus to ’hallucinate’ the details. In this paper, the authors will explore different architectures and loss functions for the task of Super-Resolution and set them in context by comparing them against the classic up-sampling methods as a baseline.

II. Related Work

In recent years, deep neural networks have pushed the frontiers of what was previously possible in the computer vision world. The same holds true for the task of super-resolution, where deep neural networks such as *SRCNN: Image Super-Resolution using Deep Convolutional Networks* [2] outperform classical algorithms.

The astoundingly good results that current state-of-the-art methods produce are exemplified by CAR [8] or SwinIR [6] that make use of either transformer architectures or smart (re-)sampling. Other notable methods include *Adjusted Anchored Neighborhood Regression for Fast Super-Resolution (A+)* [9], *Single-image super-resolution using sparse regression and natural image prior* [5] or *Super-Resolution From a Single Image* [3].

III. Approach

A. Network Architectures

In essence, there are four CNN frameworks to perform the learned upsampling operation:

- pre-upsampling (see Fig.1)
- post-upsampling
- progressive uosampling
- iterative up-and-down-sampling

The scope of this project was limited to fully-supervised pre-upsampling methods.

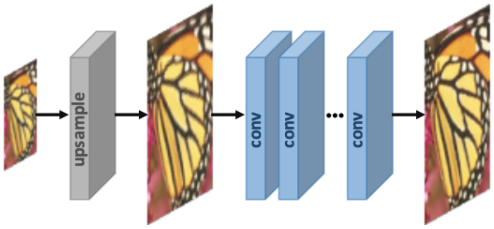


Fig. 1: Pre-Upsamplling Frameworks [10]

1) SRCNN or Super-Resolution-Convolutional-Neural-Network:

SRCNN is a representative of the class of learning-based super-resolution methods. SRCNN together with VDSR can be classified as pre-upsampling methods as both first upsample the low resolution image with, for example, bicubic interpolation before feeding the images into the convolutional neural networks. The CNNs carry the interpolated input images to the end layer and reconstruct - or hallucinate - additional (often high-frequency) information on top of the input image.

The architecture of the SRCNN model is shown in Fig.2. It implements three convolutional layers with feature maps of 64 and 32 in between. The SRCNN model constructs a best-guess of the HR image based on the upsampled LR image in an end-to-end manner. Compared to most CNN for Super Resolution, the SRCNN with its merely 3 layers is a rather shallow deep neural network. To put this into perspective, VDSR in its original form counts 20 layers.

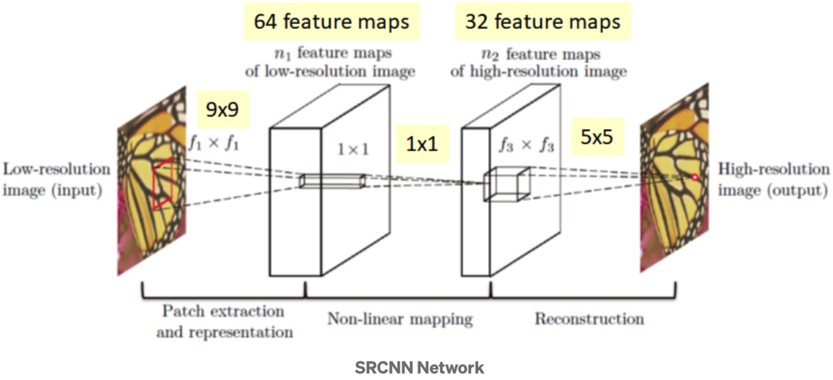

64 feature maps
n_1 feature maps of low-resolution image
32 feature maps
n_2 feature maps of high-resolution image
9x9
$f_1 \times f_1$
1 × 1
1x1
$f_3 \times f_3$
5x5
Low-resolution image (input)
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction
SRCNN Network

Fig. 2: SRCNN model from original paper [2]

2) *VDSR or Very-Deep-Super-Resolution:*

VDSR, which stands for Very Deep Super Resolution, is a deep neural network with variable layer depth D, which was introduced in 2016 by Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee [4]. This method uses a very deep convolutional network because it was found that increasing the network depth D improves the accuracy accordingly.

The pre-upsampled input image (ILR) passes through (D-1) layers, each consisting of a convolutional layer and a ReLU activation layer, followed by a single, final convolutional layer without activation. The output of this last convolutional layer will in the following be called the 'residual'. To produce the final output, the residual is added to the ILR (interpolated low resolution) input image. The convolutional layers that form the backbone of the model have filter sizes of 3x3, stride of 1 and a constant feature size (number of filters) of 64. The original VDSR model had D=20 weight layers. The way in which VDSR work is that contextual information is extracted over large image regions in an efficient way by streaming small filters multiple times in a deep network structure. VDSR offers a simple, yet effective training procedure.

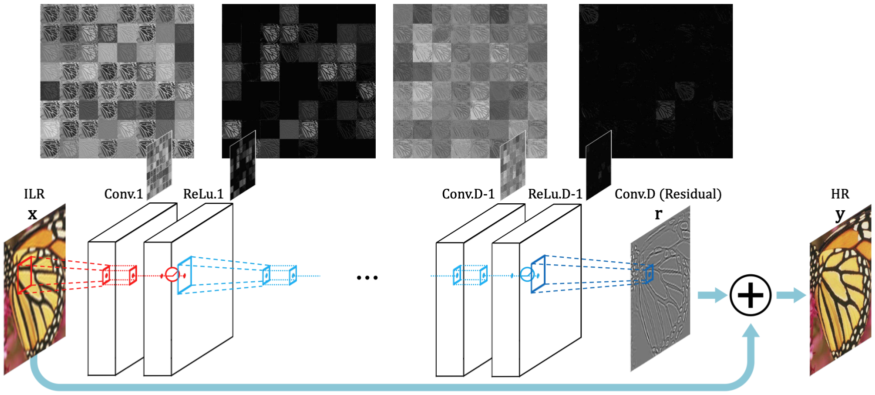


Fig. 3: VDSR model from [4]

Network	Trainable Parameters	Upsampling	other characteristics
SRCNN	69.251	Pre-Upsampling	
VDSR (d=5)	187.776	Pre-Upsampling	residual

Like many SR methods, SRCNN and VDSR are trained specifically and exclusively for a single scale factor and they are supposed to work expertly only for that specific upsampling scale factor. Therefore, demand for a new scaling factor requires a newly trained model.

B. Training

1) Dataset: Datasets for SR are widely available and easy to generate. In contrast to other Computer Vision tasks, such as Segmentation or Tracking, where human input is indispensable for generating ground-truth masks, it is relatively easy and computationally inexpensive to create HR-LR image pairs for training by down-sampling available HR images.

Set 5 [1]: The Set5 Dataset consists of only five images - one for each category in {baby, bird, butterfly, head, woman}. These images are available at scaling factors of 2x, 4x, 3x and 8x. Set5 itself is too small to train on the five images only. Nevertheless, Set5 can be used in combination with e.g. Set14 or 91 Images as a supplementary evaluation set. The currently best performing model on Set5 at 4x upscaling is SwinIR [6]

BSD100 or The Berkeley Segmentation Dataset and Benchmark [7]: The BSD100 is a widely-used dataset for image upsampling (super-resolution) or de-noising. It contains 100 images of natural objects such as plants or people.

Dataset	Number of Images	avg. Resolution	Upscaling Factor	Channels Info
Set5	5	(313, 336)	2x, 3x, 4x	1 channel (black and white)
91-Images	91	(264, 204)	2x, 3x, 4x	1 channel (black and white)
BSD100	100	(432, 367)	2x, 3x, 4x	3 channel (RGB)

2) Data Augmentation: Prior to training, the images within the datasets had to be standardized to a uniform size. Because rescaling images comes with having to interpolate between pixels, it was not clear what implications the rescaling-operation on the training images could have on the final results. Therefore, we favoured cropping the images instead of resizing them. As a result, all images within one dataset were cropped to the size of the smallest image within the dataset. Data augmenting transforms (horizontal and vertical flips) were employed.

3) Loss Functions: The task of the loss function is to accurately quantify the reconstruction error of the neural network. The loss function guides the learning process. The loss functions that were considered for this project were the pixelwise L1 loss, the MSE loss (also known as the L2 loss), and a variant of the L1 loss, the Charbonnier Loss. Here, I and $\hat{I}$ denote, respectively, the target high-resolution image and the generated high-resolution image.

MSE (Mean-Squared Error or L2-Loss):

$$L_{MSE} = \frac{1}{H * W * C} \sum_{i,j,k} |\hat{I}_{i,j,k} - I_{i,j,k}|^2 \tag{1}$$

L1-Loss:

$$L_{L1} = \frac{1}{H * W * C} \sum_{i,j,k} |\hat{I}_{i,j,k} - I_{i,j,k}| \tag{2}$$

Charbonnier Loss:

$$L_{L1} = \frac{1}{H * W * C} \sum_{i,j,k} \sqrt{(\hat{I}_{i,j,k} - I_{i,j,k})^2 + \epsilon^2} \tag{3}$$

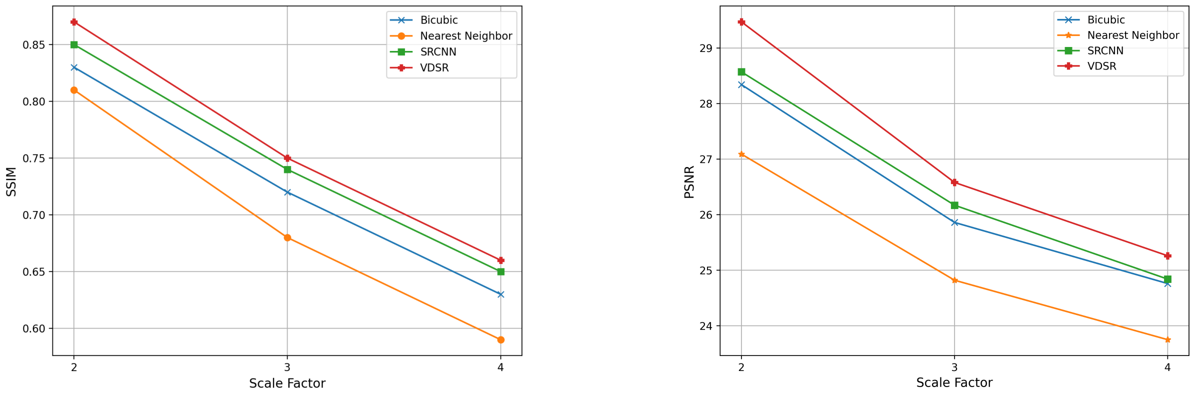
with a small constant $\epsilon \leq 1e-3$ for numeric stability.

It has been shown experimentally [10] that the MSE loss generally creates over-smooth results as the squared term heavily penalizes large differences between I and $\hat{I}$ but tolerates small deviations. The experiments were therefore carried out with the L1 Loss, which does not suffer from the same disadvantages. Besides pixel-wise loss functions, there are, among others, the option to employ the content loss, an adversarial loss or the total variation loss.

4) Evaluation Metrics: To assess the quality of the upsampled image, a benchmark that considers different perceptual qualities is needed. The authors propose the use of PSNR (Peak-Signal-To-Noise-Ratio) and SSIM (Structural Similiarity Index). As each score focuses on a different visual quality of the image, consistency between these two metrics is not necessarily a given, both scores were reported during training and evaluation.

PSNR (Peak Signal-To-Noise Ratio):

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{L^2}{\frac{1}{N}\sum_i (I(i) - \hat{I}(i)^2)}\right) = 10 \cdot \log_{10}\left(\frac{L^2}{\text{MSE}}\right) \tag{4}$$



(a) SSIM

(b) PSNR

Fig. 4: Metrics on BSD100

PSNR is a logarithmic measure which is defined in terms of the maximum pixel value L and the MSE error between two images. If the L2 loss is chosen as the driving loss function, then PSNE is maximized directly. In reality, PSNR performs poorly on real scenes. However, it is the most widely used evaluation criterion for super-resolution and for evaluating lossy transforms in general.

SSIM (Structural Similarity Index Measure): SSIM was used as an adjunct to PSNR for the evaluation of the final results. SSIM is an index of the similarity of structural attributes between two images. It is based on the luminance and contrast info in the images. This is why SSIM outperforms PSNR as a measure of brightness and structure similarities.

IV. Experiments & Comparison

A. Results of the experiments with learning based methods

Due to hardware limitations, the depth (= number of layers) of the VDSR model was reduced to five. This was expected to deteriorate reconstruction quality considerably. Nethertheless, VDSR is the clear winner on all scale factors and datasets.

Results on the BSD100 Dataset:

Scale Factor	Metric	Bicubic	Nearest Neighbor	SRCNN	VDSR
2	PSNR [dB]	28.34	27.09	28.57	29.47
2	SSIM	0.83	0.81	0.85	0.87
3	PSNR [dB]	25.86	24.82	26.17	26.58
3	SSIM	0.72	0.68	0.74	0.75
4	PSNR [dB]	24.76	23.75	24.84	25.26
4	SSIM	0.63	0.59	0.65	0.66

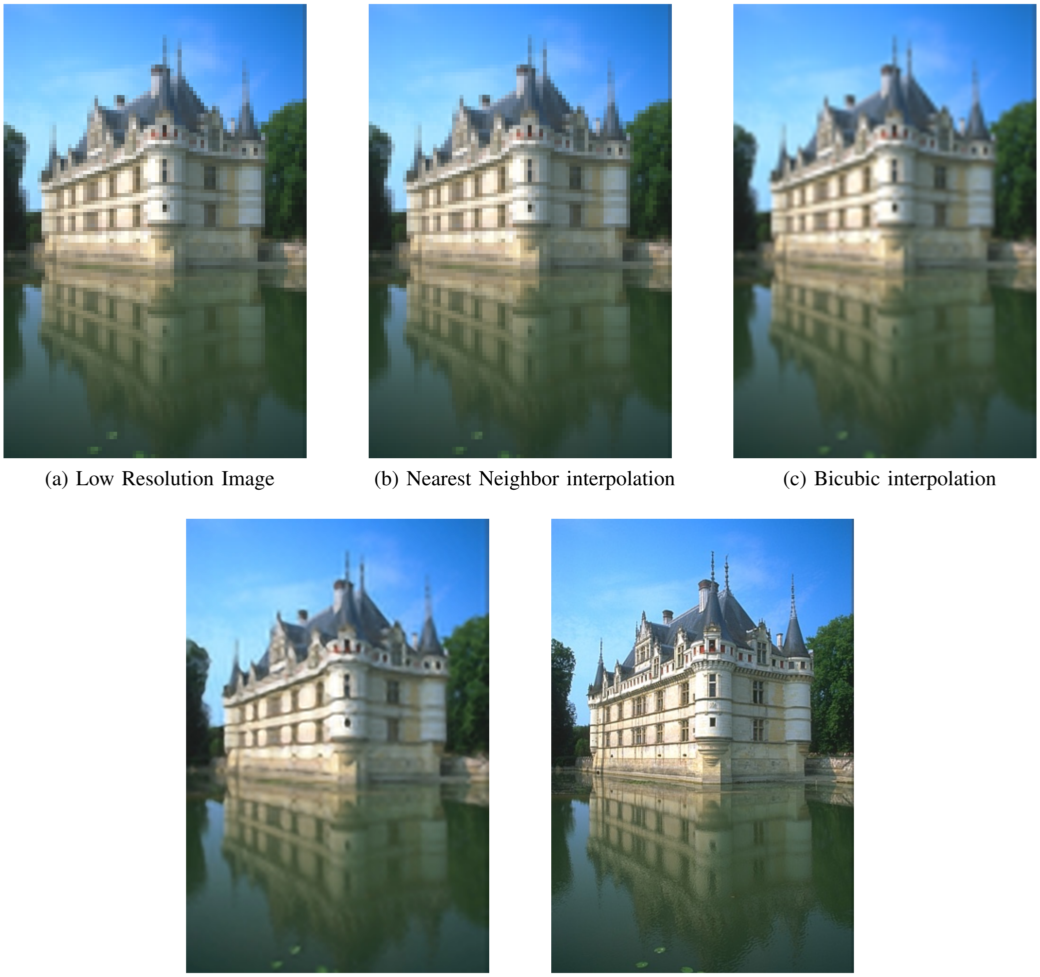
TABLE I: List of Results on BSD100

Results on 91-Images:

Scale Factor	Metric	Bicubic	SRCNN	VDSR
2	PSNR [dB]	34.61	36.94	37.74
2	SSIM	0.88	0.92	0.93
3	PSNR [dB]	31.50	33.14	33.86
3	SSIM	0.77	0.82	0.84
4	PSNR [dB]	29.31	30.62	31.21
4	SSIM	0.67	0.73	0.75

TABLE II: List of Results on 91-Images

It is evident from Tables II and I, that both SRCNN and VDSR outperform the non-parametric bicubic and nearest-neighbor upsampling in all scenarios, irrespective of the upsampling factor or the dataset. A clear hierarchy of performance can be established:



(a) Low Resolution Image (b) Nearest Neighbor interpolation (c) Bicubic interpolation

(d) SRCNN (e) HR Image

Fig. 5: Comparison: Upsampling Methods on BSD100

Bicubic interpolation is consistently worse than SRCNN and VDSR on both datasets at upsampling factors of 2x, 3x and 4x. The structural similarity index measure follows the same trend. The difference for the BSD100 dataset between the bicubic interpolation and SRCNN is independent of the scale factor: it is worse by $\Delta\mathrm{SSIM} = 0.02$ than SRCNN. In comparison to VDSR, the difference is between 0.03 and 0.04 regardless of the scale factor. As was to be expected, a lower scaling factor results in higher PSNR and SSIM scores across all methods.

V. Conclusion

In this paper the authors explored different architectures and loss functions for the task of up-sampling low-resolution images to meaningful high-hesolution images. The experiments have shown that the classic up-sampling method bicubic interpolation is outperformed by both SRCNN and VDSR. Furthermore, VDSR yields better results than SRCNN which indicates that using very deep neural networks that consequently have many thousands or millians of trainable parameters enhances the quality of the generated HR image. It is also noticeable that VDSR outputs not only the best results but also reaches the best results in a short time by a much faster learning rate than SRCNN.

References

[1] Marco Bevilacqua, A. Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single image super-resolution based on nonnegative neighbor embedding. 09 2012.

[2] Kaiming He Xiaoou Tang Chao Dong, Chen Change Loy. Image super-resolution using deep convolutional networks. In *Learning a Deep Convolutional Network for Image Super-Resolution*, 2015.

[3] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 349–356, 2009.

[4] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks, 2015.

[5] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.

[6] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *IEEE International Conference on Computer Vision Workshops*, 2021.

[7] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[8] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020.

[9] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. volume 9006, pages 111–126, 04 2015.

[10] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. Deep learning for image super-resolution: A survey, 2019.

ACKNOWLEDGMENT

The authors would like to thank Prof. Rother for the engaging lecture.